

인공지능의 하극상과 그에 대처하는 바람직한 프로그래머의 자세

박선호

October 26, 2022

Abstract

인공지능이 빠르게 발전하며 프로그래머마저 인공지능으로 대체될 수 있다는 주장도 있는데 근미래에 가능할 것이라고는 보기 힘들다. 그 이유로는 많이 사용되는 인공지능망 방식이 근본적으로 논리적 사고를 하기 힘들데, 이 논리적 사고력은 프로그래밍에서 반드시 필요하기 때문이다. 만약 기술 발전으로 가능하게 되더라도 최종 검수 단계나 인공지능에게 의사를 전하는 과정에서 전문적인 프로그래밍 지식과 능력은 필요하다. 비록 인공지능이 프로그래머를 대체하지는 못하더라도, 인공지능이 프로그래머의 조력자인 것처럼 사용될 것이므로 이를 잘 다룰 수 있는 능력이 앞으로의 프로그래머에게 필요할 것이다.

인공지능 기술이 발전을 거듭하며 예전에는 불가능할 것이라고 생각되었던 일까지 자동으로 할 수 있게 됨에 따라 몇몇 직업들은 사라질 위기에 처해 있다. 예를 들어, 최근 깃헙(Github)에서 개발한 코파일럿(Copilot)이라는 인공지능은 프로그래머의 의도에 맞게 적절한 코드를 자동으로 생성해주는데, 이 기능을 편리하게 잘 사용하는 사람도 있는 반면 일각에서는 이 인공지능 서비스로 인해 장차 프로그래머라는 직업조차 인공지능에 대체되어 버릴 수 있다며 양날의 검이라는 경고의 목소리도 나온다. 심지어 단순 노동이나 운전과 관련된 직업과 같은 경우는 인공지능에 의해 대체될 수 있다는 사실은 계속 예고되고 있다. 이 상황에서 여러분은 프로그래머가 인공지능으로 대체될 것이라고 생각하는가? 만약 프로그래머가 인공지능으로 대체된다면 이는 곧 인공지능이 하극상을 일으켜 자신들을 개발한 프로그래머의 밥그릇을 빼앗는 일이 생긴다는 얘기인데, 이러한 예상으로 인해 현직 프로그래머 및 프로그래머 지망생 중 불안해하는 사람이 상당수 있을 것이다. 그러한 사람들에게 이 글은 희망적인 이야기가 될 것이다. 근미래에 프로그래머가 전부 인공지능으로 대체되긴 힘들다고 생각할 수 있는 몇 가지 근거를 공유하고 함께 생각해보고자 한다.

가장 큰 이유로, 현재 유행하고 있는 인공지능망 모델은 근본적으로 논리적 사고가 하기 힘들다는 점을 들 수 있다. 2011년에 심리학자인 대니얼 카너먼(Daniel Kahneman)이 쓴 책인 ‘생각에 대한 생각’ (Thinking, Fast and Slow)에서는 사고 모드를 시스템 1과 시스템 2로 분류했다. 시스템 1은 직관적이고 빠른 사고로 직관이 크게 작용하는 사고 모드이며, 시스템 2는 시스템 1로 해결이 안되는 복잡한 문제를 해결하는 데 사용되며 논리적이고 통계적인 사고가 가능하지만 시간이 오래 걸리는 사고 모드이다. 여기서 주목해야 하는 점은 CACM의 2022년 10월 기사에 실린 텍사스 대학교의 교수와 진행한 인터뷰에서도 나와 있듯¹, 현재 인공지능은 시스템 1의 사고가 가능할 뿐 시스템 2의 사고는 힘들다는 점이다. 현재의 인공지능은 인공지능망 위에서 수많은 데이터를 사용하여 학습 과정이 진행되는데, 여기서 학습 과정이라는 것은 통계학과 미적분학을 이용하여 데이터로부터 특징을 자동적으로 추출하여 모델의 내부 값들을 조정하는 것을 의미한다. 이 특징을 추출하는 과정 자체가 논리적 사고를 가능하게 한다기 보다는 직관력을 기른다고 보는 것이 타당하며 이로부터 인공지능망은 시스템 1 사고 모드를 위한 모델임을 이해할 수 있다. 이때까지 인공지능이 놀라운 성과를 보여준 이미지 분류, 음성 인식, 심지어는 자율주행 같은 기술들 모두 직관력이 중요한 분야임을 확인할 수 있다.

하지만 프로그래밍의 경우에는 시스템 1의 사고 능력에 못지 않게 시스템 2의 사고 능력인 논리적 사고력도 중요하다는 사실은 부정할 수 없다. 과거부터 현재까지 프로그램의 취약점이나 버그로 인해 로켓이 추락하거나 사용자가 보안 위험에 처하는 등 여러 문제점이 있어 왔고, 이러한 버그들은 프로그래밍의 단계에서 주로 개발자의 부주의로부터 비롯한다. 즉 개발자의 시스템 2 사고력이 부족해서 생긴 문제인데, 현재의 인공지능은 시스템 2의 사고력이 부족하므로 코드에 취약점을 많이 만들어낼 가능성이 있다. 깃헙의 코파일럿만 보더라도 의미 없는 결과를 만들어내거나 취약점이 담긴 코드를 만들어내는 사례를 쉽사리 확인할 수 있다.

¹Don Monroe, “Neurosymbolic AI”, Communications of the ACM, October 2022, Vol. 65 No. 10, Pages 11-13

그러므로 인공지능이 시스템 2의 사고 능력을 가질 수 있어야 하는데 이 문제가 쉽사리 해결되리라고는 생각하기 어렵다. 현재까지 알려진 방식으로는 프로그래밍 언어 이론을 이용하여 정확하게 프로그램을 만들어 내는 방식과 인공지능을 이용하여 통계적으로 프로그램을 만들어 내는 방식이 알려져 있다. 프로그래밍 언어 이론을 이용하는 방식은 보통 굉장히 큰 탐색 공간(search space) 안에서 적절한 알고리즘을 이용하여 프로그래머의 의도에 맞는 프로그램을 찾는데, 현재로써는 비교적 짧은 코드를 합성할 때도 굉장히 많은 시간이 필요하다는 문제점이 있다. 인공지능을 이용하는 방식은 시스템 2의 사고 능력을 학습시킬 수 있어야 정확한 코드가 생성된다는 사실을 보장할 수 있는데, 많은 연구자들이 연구하고 있음에도 불구하고 획기적인 돌파구가 나오지 않고 있다. 역사적으로 인공지능은 몇차례 한계에 부딪혀서 암흑기에 빠졌는데, 시스템 2의 사고 능력의 한계 때문에 새로운 암흑기에 빠지게 될 지는 아무도 모르는 일이다. 그러므로 인공지능이 프로그래머를 완전히 대체하는 미래는 당장에 기대하기 어렵다고 생각한다.

여기까지 읽었을 때, 아마 프로그래밍 언어 이론과 인공지능을 적절히 융합하여 정확한 코드를 빠르게 만들어내는 서비스가 나타날 것이라며 이 글에 반박하려 할지도 모른다. 하지만 만약 어떠한 방식으로든 인공지능이 프로그래밍을 완전히 할 수 있게 되더라도 프로그래머는 완전히 대체되기 힘들다고 생각한다. 그 이유는 사람의 의도에 맞게 프로그램이 만들어졌는지 점검하는 과정은 사람밖에 할 수 없기 때문이다. 현재도 개발자들은 협업 과정에서 의사소통의 부재로 인한 여러 문제점에 직면하는데, 인공지능과도 완벽하게 의사가 전달되지 못할 경우 그러한 문제가 발생할 수 있으므로 프로그래머가 최종 단계에 검수하는 과정을 거쳐야 한다. 또한, 하나의 문제를 해결하는 데도 서로 다른 알고리즘이 존재하고 각각의 알고리즘마다 실행 속도나 메모리 사용량 측면에서의 장단점이 있는데, 인공지능에게 어떠한 알고리즘을 사용해야 하는지 지시하거나 직접 최적화를 하는 것은 여전히 사람의 몫이 될 것이다. 그러므로 인공지능이 코딩을 굉장히 잘하게 되더라도 전문적인 프로그래머는 계속 필요할 것이라고 예상할 수 있다.

이렇듯 프로그래머가 대체되지 않을 것이라고 했지만, 프로그래밍에 변화가 거의 없을 것이라는 얘기는 아니다. 지금도 귀찮은 작업에 코파일럿의 도움을 받는 프로그래머들이 꽤 있으며, 앞으로는 생산성을 위해 이러한 식으로 인공지능이 코딩을 보조하는 방식으로 존재할 가능성이 높다고 본다. 이러한 변화하는 트렌드에 맞춰서 앞으로는 타인 뿐만 아니라 인공지능과도 협업을 해야 하게 될 것이고, 이를 위해서는 효율적인 모듈화를 통해 인공지능에게 일거리를 알맞게 떼어 내 줄 수 있는 능력이 필요할 것으로 보인다. 또한 인공지능이 생성한 코드에 버그가 있으면 찾기 힘들므로, 인공지능이 생성한 코드가 정말로 옳은 코드인지 개발 과정에서 정확하게 분석하는 능력도 상당히 중요할 것으로 예상된다. 그러므로 이러한 능력을 바탕으로, 인공지능이 프로그래머를 대체한다는 불안에서 벗어나 인공지능이라는 새로운 조력자와 함께 프로그래밍을 하며 자신의 생산성을 높이는 데 집중하는 프로그래머가 되어야 할 것이다.